

```
1  ////
2  // Name: Andrew Norton
3  // Section: CS 132 Winter 24
4  // Class Name: ANString
5  //
6  // Description: This class holds a C String, keeps count of
7  // its objects it creates, and uses overloaded operators for
8  // functionality.
9  ////
10
11 #ifndef ANSTRING_H
12 #define ANSTRING_H
13
14 #include <istream>
15
16 using namespace std;
17
18 class ANString {
19 public:
20     ANString(); // default constructor
21     ANString(const ANString& mstr); // Copy Constructor
22     ANString(const char* cstr); // cstring constructor
23     ~ANString(); // Destructor
24
25     ANString operator= (const ANString& rvalue); // Assignment operator
26     ANString operator+(const ANString& rvalue); // Plus operator
27     char operator[] (int index); // non const Index operator
28     friend istream& operator>>(istream& istrm, ANString& string) { // write ↗
29         for operator (was write function)
30             char inputWord[100];
31             if (istrm >> inputWord) {
32                 for (string.end = 0; inputWord[string.end] != '\0'; + ↗
33                     +string.end); //empty loop
34
35                 if (string.end >= string.cap)
36                 {
37                     string.cap = 20 * (string.end / 20 + 1);
38                     delete[] string.str;
39                     string.str = new char[string.cap];
40                 }
41
42                 // if (end >= cap) cap = 20 * (end/20 + 1) possibly?
43                 for (int i = 0; i <= string.end; ++i) {
44                     string.str[i] = inputWord[i];
45                 }
46             }
47             return istrm;
```

```
48     }
49
50     bool operator<(const ANString& argStr); // less than conditional      ↗
        operator
51     bool operator>(const ANString& argStr); // greater than conditional    ↗
        operator
52     bool operator==(const ANString& argStr); // equal to conditional      ↗
        operator
53
54     int length();
55     int capacity();
56     static int getCurrentCount();
57     static int getCreatedCount();
58
59     const char* c_str() const;
60
61 private:
62     char* str;
63     int end;
64     int cap;
65     static int currentCount;
66     static int createdCount;
67 };
68 ostream& operator<<(ostream& ostrm, const ANString& string);
69
70 #endif
71
72
```